

MAS101 : CODING THE MATHEMATICS - AN OOP APPROACH

April 17, 2018

Lecturer:	Seungwoo Schin	Time:	TBA
Email:	principia_12@kaist.ac.kr	Place:	TBA

Study Objectives 머신 러닝에 필요한 수학적 개념을 직접 파이썬으로 구현해보면서 익히는 것을 목표로 합니다.

Target Audience

- 선형대수학과 미적분학, 확률 및 통계의 기본적인 개념이 궁금하신 분
- 다양한 코딩 기술을 다지고 싶으신 분

Things to Learn

- 벡터, 행렬에 대한 기본적인 개념 및 연산
- 선형대수학을 활용한 문제풀이
- 수학 문제 적용에 대한 예시 찾아보기

Policy

- 본 강의는 수학 이론을 배우고, 배운 이론을 구현해보는 것으로 이루어져 있습니다.
- 실습 코드는 개설될 Github Repository에 커밋하여 공유하게 할 예정입니다.
- 실습을 하는 것이 의무는 아니지만, 가능하면 실습 문제를 해결해볼 것을 권장드립니다.

Prerequisite : 기본적인 파이썬 문법에 대한 지식이 필요하며, 중학교 수준의 수학 지식을 가지고 있어야 합니다. 또, 가능하면 git을 사용할 줄 알면 좋습니다.

Tentative Course Outline

- Week 0(before class) : Environment Settings / Python Reminder
 - Python 개발 환경 설치
 - 클래스 및 객체지향 : 문법, method, 상속
 - 문제 풀이 : 피보나치 수 계산하기, 정렬, 소인수분해
 - 소스 코드 작성, 저장, 실행 및 버전관리 기초 : Git (*Optional*)
- Week 1 : Math Basics/Implementation
 - 수학
 - * 집합(set)의 정의 및 연산
 - * 함수(function)의 정의
 - * 유명한 함수들과 그 성질 살펴보기 : sin, cos, tan, log, exp
 - 프로그래밍
 - * 객체지향적 사고 : 클래스를 이용한 집합 및 함수 구현
 - * 재귀 : 피보나치 수열 계산으로 본 재귀함수
 - * 알고리즘의 시간복잡도 표기법 및 계산
- Week 1.5 : Documentation/Version Control
 - I^AT_EX 소개
 - git 소개
- Week 2 : Linear Algebra(1)
 - 수학
 - * 벡터와 행렬의 정의 및 연산 : 크기, 사칙연산, 행렬식
 - * 직교좌표계 : 두 좌표간의 거리, 직선, 평면의 방정식
 - * 선형독립과 span
 - * 선형변환
 - * 벡터와 함수 : 함수를 element로 가지는 벡터와, 벡터를 argument로 가지는 함수
 - 프로그래밍
 - * Overloading : 행렬/벡터의 연산 구현
 - * A bit of Functional Approach : functions as values
- Week 2.5 : Support Vector Machine
 - Linearly Seperable Case
 - non-linear case : Using Kernel Function
 - Example on text classification : Linear SVM model for MNIST
- Week 3 : Linear Algebra(2)
 - 수학
 - * 행렬의 대각화
 - * 행렬의 고유값 찾기
 - * 행렬의 분해 : Singular Value Decomposition(SVD), QR Decomposition

- 프로그래밍
 - * 2주차의 행렬 클래스를 확장하여 위 내용들 구현하기
 - * 선형방정식 solver 만들기
- Week 3.5 : Various Applications of Matrix
 - wavelet을 이용한 이미지 손실 압축
 - Principal Component Analysis(PCA) 구현해보기
 - 선형계획법/게임이론 (*optional*)
- Week 4 : Calculus(1) - 미분
 - 수학
 - * 함수의 극한
 - * 미분의 정의 및 구현 : Symbolic/Numeric
 - 프로그래밍
 - * 수식 parser/계산기 구현
 - * Symbolic 미분기 구현
 - * Numeric 미분기 구현
- Week 4.5 : 정규표현식
 - Parsing이란?
 - 정규표현식이란?
 - 정규표현식 파서 만들기
- Week 5 : Calculus(2) - 적분
 - 수학
 - * 적분의 정의 : 리만 합
 - * 미분과의 관계
 - * 정적분/부정적분 : 다양한 함수의 적분 해보기
 - 프로그래밍
 - * Symbolic 적분기 구현 (부정적분, 정적분 모두에 대해서)
 - * Numeric 적분기 구현 (정적분에 대해서)
- Week 5.5 : Fourier Transform
 - Fourier Transform 이론
 - Discrete Fourier Transform 이론 및 Fast Fourier Transform 구현
 - FFT를 이용한 악기 소리 샘플 분석
- Week 6 : Calculus(3) - 다변수 미분
 - 수학
 - * 편미분
 - * 그래디언트와 활용
 - * 라그랑주 승수법
 - 프로그래밍

- * 편미분 구현
- * 그래디언트 구현
- Week 6.5 : Gradient Descent
 - Gradient Descent 소개
 - 다양한 Gradient Descent 알고리즘 구현
- Week 7 : Probability and Statistics
 - 수학
 - * 확률의 정의
 - * 조건부확률과 베이즈 정리
 - * 확률변수와 확률분포함수
 - * 정규분포
 - * Central Limit Theorem (*optional*)
 - * Bayesian Statistics
 - 프로그래밍
 - * 수치적분을 통한 확률값 계산
- Week 7.5 : Feedforward Neural Net
 - Feedforward Neural Net 이론 소개
 - Backpropagation / Gradient Descent
 - Feedforward Neural Net 구현
- Week 8 : Introduction to Machine Learning - statistical classification
 - 머신 러닝이란?
 - 머신 러닝의 분류 : supervised/unsupervised, classification/regression/clustering
 - Classification 알고리즘들 소개 : Linear Regression, NBC, LDA